**Kosárlabda Edzésnapló**

A Kosárlabda Edzésnapló projekt célja egy olyan webalkalmazás létrehozása, amely lehetővé teszi kosárlabdázók számára, hogy nyomon követhessék edzéseiket, rögzítsék a gyakorlatokat, edzésidőt és megjegyzéseket. A felhasználóbarát felület biztosítja, hogy az adatok könnyen kezelhetők és áttekinthetők legyenek.

**Backend – Laravel:**

* Code First logika
* POST végpont: (Egy végpont (/api/edzesek) az adatok mentésére, ellenőrzi a kapott adatokat: játékos neve, dátum, edzés időtartama, gyakorlatok JSON formátumban, megjegyzések. Hibás adatok esetén validációs hibaüzenetet ad vissza.)
* GET végpontok: (Egy végpont az összes edzés lekérésére (/api/edzesek).)
  + Legalább három szűrt lekérdezés:
  + Edzések lekérése játékos szerint
  + Edzések lekérése dátum szerint
  + Edzések lekérése időtartam szerint

**Fronted – React:**

* Reszponzív kialakítás
* Adatok megjelenítése
* Layout alkalmazása

**Fejlesztés 10.31.**

Először létrehoztam a Laravel projektet a gépemen, beállítottam az adatbázist a .env fájlban, és létrehoztam az Edzes modellt a hozzá tartozó migrációval. A migrációban definiáltam az edzéshez szükséges mezőket, mint például a játékos neve, dátum, edzés típusa, időtartam, gyakorlatok és megjegyzés.

A fejlesztés során szembesültem néhány problémával. Eleinte nem tudtam futtatni az artisan parancsot, mert nem a projekt gyökérmappájában álltam a terminálban, ezt a problémát a megfelelő mappába navigálással oldottam meg. Később SQL hibát kaptam, mert a modell egy nem létező táblát keresett, amit úgy oldottam meg, hogy a migrációt lefuttattam, és a modellben expliciten megadtam a táblanevet.

Miután a backend működött, létrehoztam az API végpontokat. Egy GET végpont az összes edzést adja vissza JSON formátumban, míg a POST végpont lehetővé teszi új edzések felvitelét. Készítettem további szűrő végpontokat is, amelyek a játékos neve, az edzés dátuma vagy az időtartam alapján listáznak adatokat.

Ezután a React frontend fejlesztésére tértem át. Létrehoztam a kosar-frontend projektet, majd készítettem két fő komponenst: az EdzesLista, ami megjeleníti az összes edzést táblázatban, és az EdzesForm, ami lehetővé teszi új edzések felvitelét. A gyakorlatok mezőt JSON formátumban tároltam, amit a frontend oldalon alakítottam tömbbé a megjelenítéshez.

A fejlesztés jelenlegi állapotában a rendszer teljesen működőképes. A React frontend megjeleníti az összes edzést, a POST végpont segítségével új edzéseket tudok felvinni, és az adatok helyesen jelennek meg. Minden korábbi hibát, például az artisan parancs problémáját és az adatbázis tábla nevét érintő hibákat sikerült megoldani.

**Fejlesztés 11.01.**

A projekt során továbbra is a Code first logikát alkalmaztam a Laravel backendnél. Az Edzes modell létrehozása után a migráció futtatásával a táblát közvetlenül a kódból definiáltam, így a backend teljesen megfelel a code first elvárásnak.

Az API végpontoknál biztosítottam, hogy a POST végpont ellenőrizze a kapott adatokat, így csak érvényes adatok kerülhetnek az adatbázisba. A GET végpontok segítségével az összes edzés lekérhető, valamint készítettem szűrő végpontokat, amelyek a játékos neve, az edzés dátuma vagy az időtartam szerint listáznak adatokat. Ezek a szűrők növelik a rendszer használhatóságát és rugalmasságát.

A frontend fejlesztésénél a React keretrendszert használtam. A projekt frontend mappában található, és két fő komponenst tartalmaz: az EdzesLista a táblázatos megjelenítésért felel, az EdzesForm pedig új edzések felvitelét teszi lehetővé. A gyakorlatok mező JSON formátumban kerül mentésre a backendbe, a frontend oldalon pedig tömbbé alakítva jelenik meg. Az oldal teljesen reszponzív, így a különböző képernyőméretekhez alkalmazkodik.

Fejlesztés közben újabb problémákkal találkoztam. Az artisan parancs kezdetben nem futott, mert nem a projekt gyökérmappájában voltam, ezt a terminálban történő helyes navigálással oldottam meg. Később a modell egy nem létező adatbázis táblát keresett, amit a migráció lefuttatásával és a modellben a táblanév explicit megadásával javítottam. Az npm parancsok futtatásakor előfordult, hogy a frontend mappa hiányzott vagy rossz helyen volt; ezt a create-react-app futtatásával és a mappa létrehozásával oldottam meg.

Jelenleg a rendszer teljesen működőképes. A React frontend megjeleníti az összes edzést, a POST végpont segítségével új edzések vihetők fel, és az adatok helyesen jelennek meg. Minden korábban felmerült hibát sikerült megoldani.

A projekt további fejlesztéséhez tervezem a layout finomítását, például egységes fejléc és szépített kártyák hozzáadását, valamint a szűrő végpontokhoz kapcsolódó frontend szűrő komponensek elkészítését, hogy a felhasználók könnyen tudjanak keresni a játékosok vagy az edzés típusok szerint.

Összességében a projekt bemutatja a Laravel és React integrációját, a backend és frontend működését, a POST és GET végpontok helyes kezelését, a reszponzív felhasználói felületet, valamint azt, hogy a fejlesztés során felmerült hibákat hogyan sikerült megoldani.

**Segédanyagok, melyek segítettek a munkámban:**

<https://www.youtube.com/playlist?list=PLVyg_FEqWfkJm8fTwumN3eYemdqnY6Zh8>

<https://react.dev>

<https://laravel.com>

<https://www.w3schools.com/Php/func_string_ucfirst.asp>

<https://www.youtube.com/@csabakiss7940>